# Dynamical Adaptation in ORNs

Srinivas Gorur-Shandilya

January 30, 2014

## Filter Estimation

How do we best calculate the filters? A summary of different ways of estimating the filters and results from each.

### Contents



### Overview of different regularisation methods.

The filter $K$ is given by

$$\hat{C} * K = s' * f$$

where $\hat{C}$ is the regularised covariance matrix $C$, $s$ is a $N$ x $M$ matrix of the time-shifted stimulus and $f$ is the response vector. $N$ is the filter length, and $M$ is the length of the stimulus and response vectors - $N$.

$C$ is the unscaled covariance matrix and is given by:

$$C = s^T * s$$

can be regularised by different means:

1) Carlotta regularises $C$ using:

$$\hat{C} = C + \hat{r}I$$

2) and Damon suggested:

$$\hat{C} = \frac{(C + \hat{r}I) * \mathrm{tr}(C)}{(\mathrm{tr}(C) + \hat{r}N)}$$

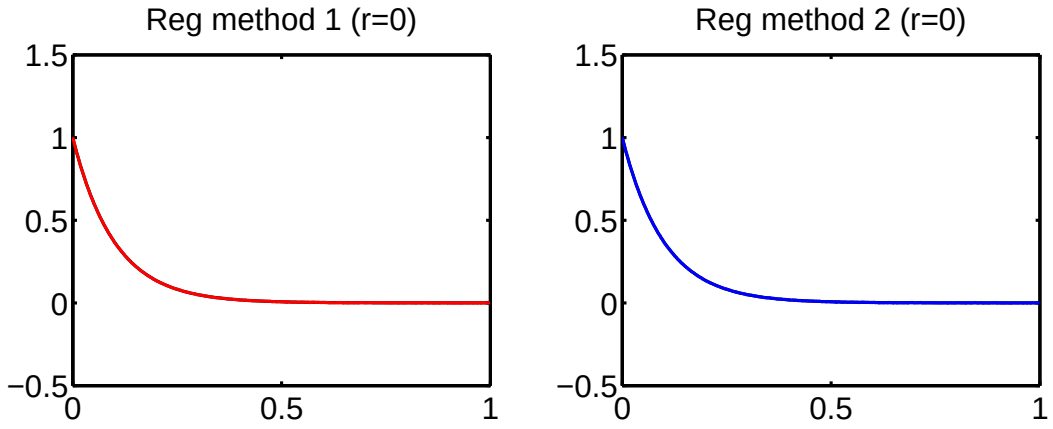
where $I$ is the identity matrix and $r$ is a free parameter called the regularisation factor that suppresses the high-frequency components of $K$. $\hat{r}$ is obtained by scaling $r$ by $\mu$, the mean of the Eigenvalues of the covariance matrix $C$. $s$ is the stimulus vector (e.g. the PID) and $f$ is the response vector (here, the firing rate of the ORN). In practice, $K$ is estimated by a left matrix division:

$$K = C \setminus (s' * f)$$

## Synthetic Data with No regularisation

Synthetic data is prepared using Gaussian random inputs and an exponential filter, and the output is the convolution of the input with the exponential filter, with 10% noise.

The following figure shows filters estimated using the three methods shown above, with zero regularisation. The actual filter is in black, and each method is coloured differently.



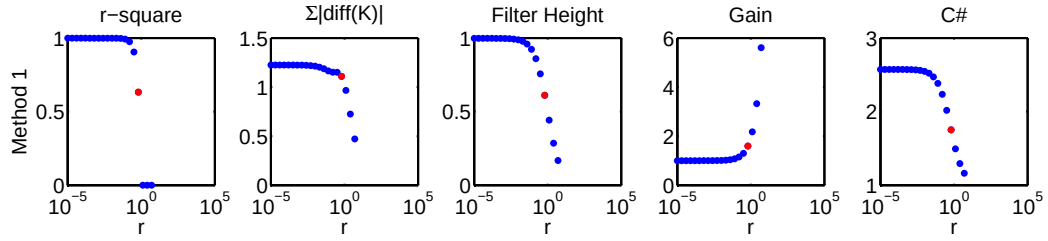

## Synthetic Data: effect of regularisation

what is the effect of increasing $r$, the regularisation parameter on each of these methods? Here we systematically vary $r$, and pick the best filter that minimises

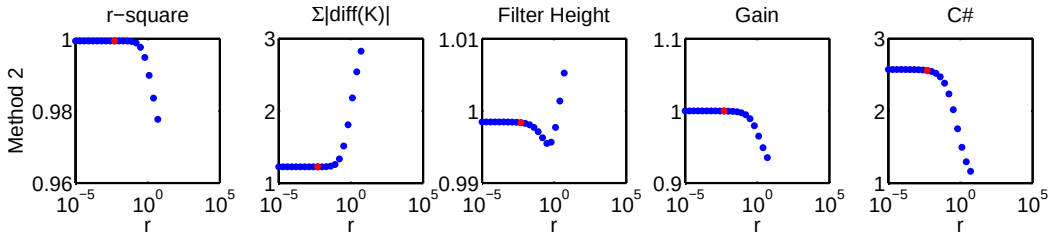
$$\arg\min_r \left\| \frac{e(r) - 1}{1}, \frac{S(r) - \min(S)}{\min(S)}, \frac{|m(r) - 1|}{1} \right\|_\infty$$

where $e$ is the coefficient of determination (r-square) of the prediction w.r.t to the actual data, $S$ is the sum of absolute values of the derivative of the filter $K$, and $m$ is the slope of the best fit of the prediction to the data. Minimising ($e$ - 1) minimises the error of the fit. Minimising $S$ minimises the high-frequency components in the filter, and prevents over-fitting. We also want the slope $m$ to be as close to 1 as possible. If $S$ has a minimum, that value of $r$ is automatically chosen.

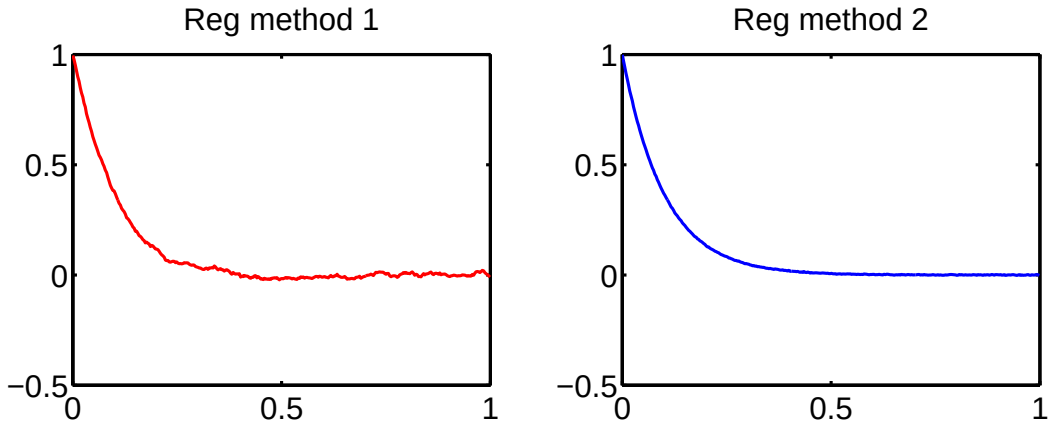
In each of the plots below, the variation of error, filter sum, filter height, gain, and condition number of $C$ is show with $r$, the regularisation factor, in units of $\mu$. The error is calculated from the prediction to the actual output. The filter sum is the sum of the absolute values of the filter. Very high values here indicate the filter dominated by high-frequency components. The filter height is the peak of the filter. The gain is the slope of the best fit line of the data to the prediction. The condition number is the ratio of the largest to the smallest eigenvalue of $C$. Values of the condition number close to 1 mean that the matrix is easier to invert. The figure below shows how varying $r$ affects reg method 1:




the figure below shows how varying $r$ affects reg method 2:



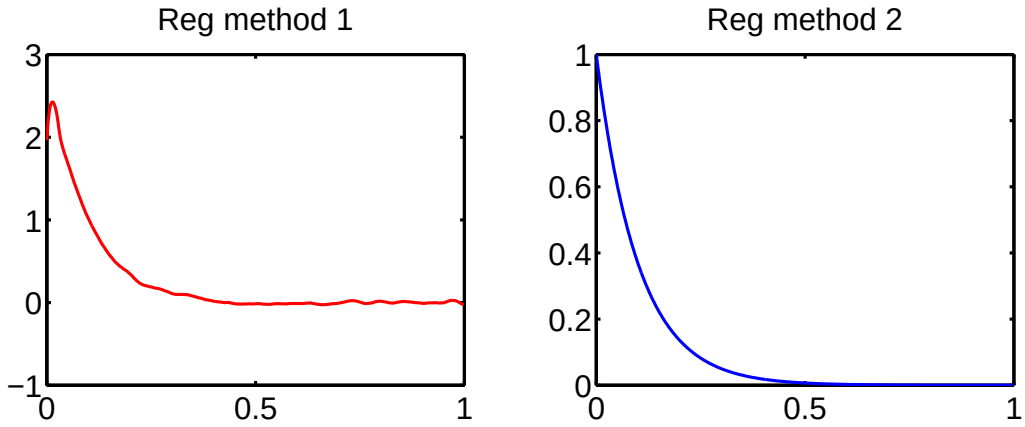

The best filters are shown below for the two reg. methods:



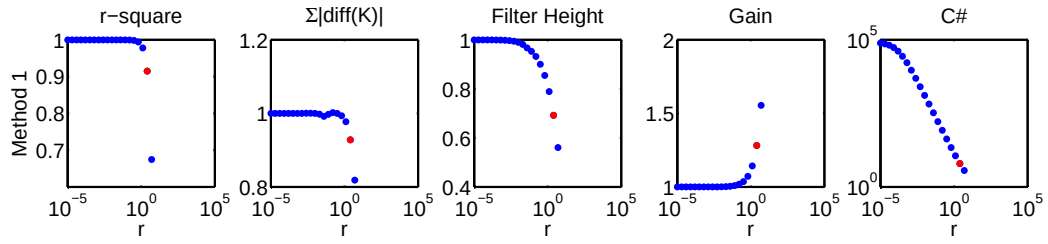

## Synthetic Data 2: effect of regularisation

We now create a new synthetic dataset, identical to the old one, except we filter the white noise input with some boxcar filter to remove high frequency components from the input. We find the best filters as before, and look at how regularisation parameter affects the choice of filter.

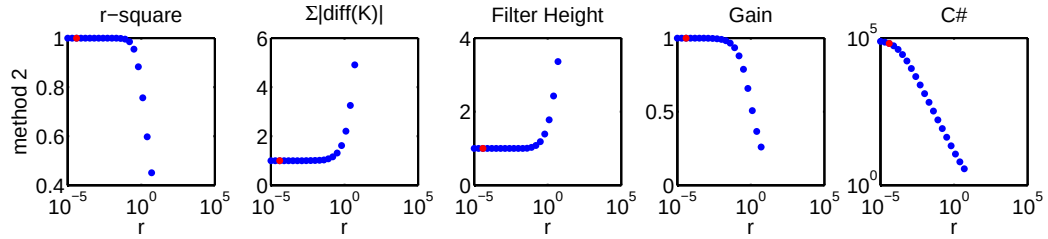
The best filters are shown below for the two reg. methods:




the figure below shows how varying $r$ affects reg method 1:

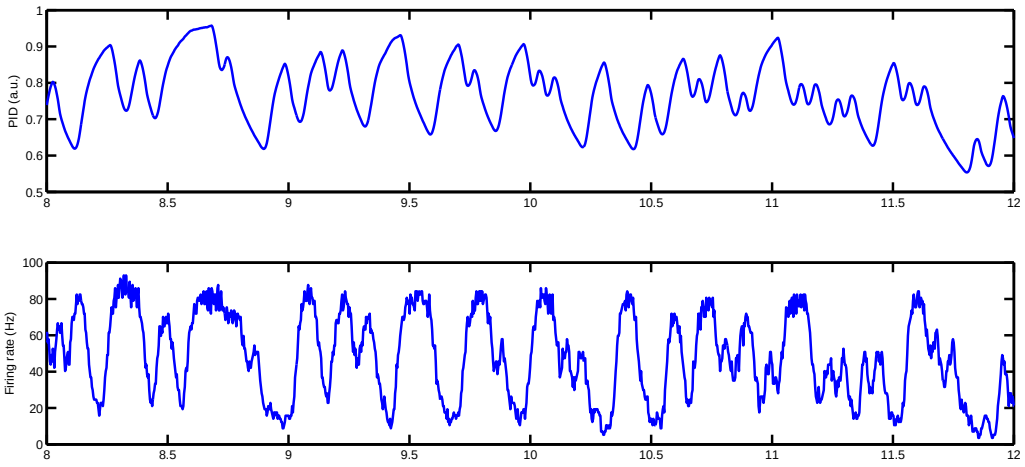



the figure below shows how varying $r$ affects reg method 2:

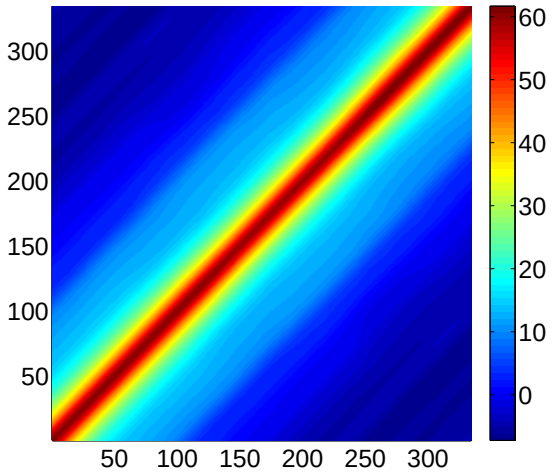



## Real Data: effect of regularisation

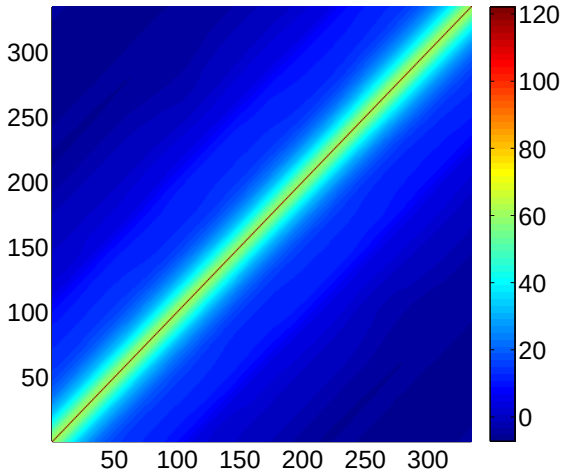
We use PID traces as the input, and the firing rate of a ORN as the output. The data looks like this:



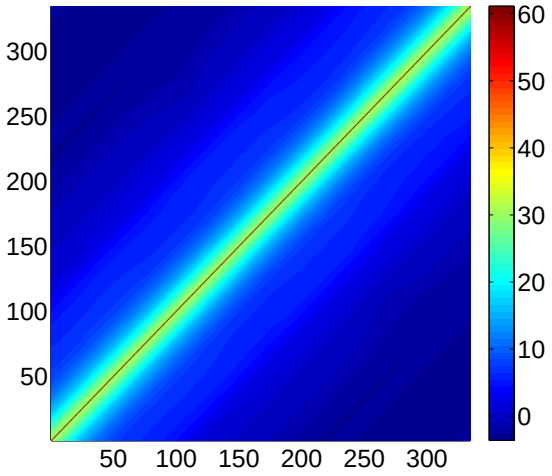

The covariance of the input matrix $C$ looks like:




Regularisation adds a large diagonal matrix to this, in effect making it look more like a diagonal matrix. Here, the mean of the eigenvalues of $C$ has been added to the diagonal. This is essentially what Method 1 does.

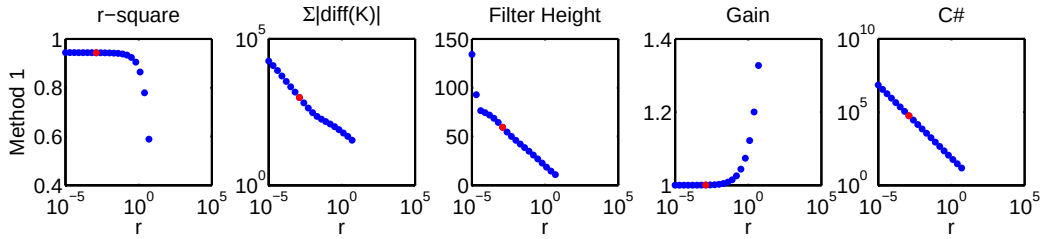



The effect of regularisation using method 2 is shown on the correlation matrix below. The same $r$ is used, i.e., the mean of the eigenvalues of $C$.

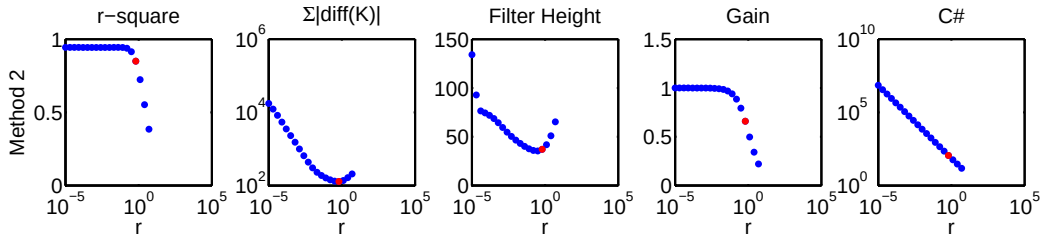



Once again, we repeat the filter extraction.

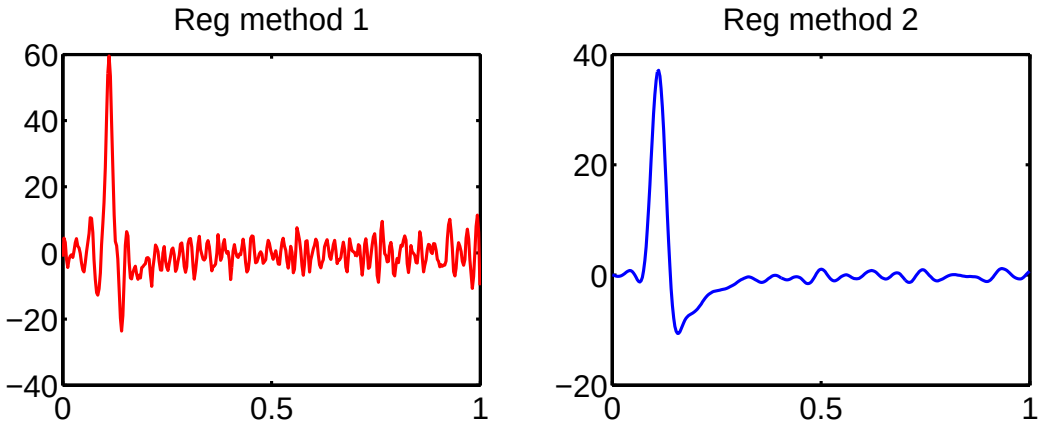
the figure below shows how varying $r$ affects reg method 1:



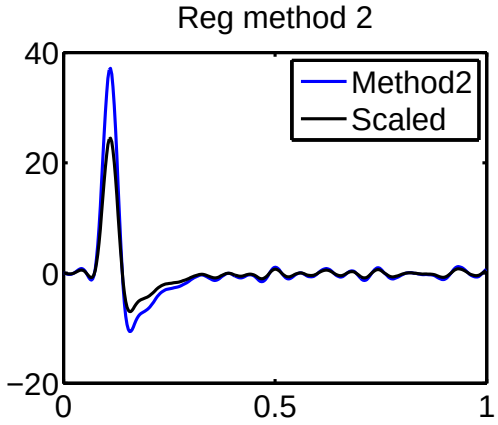

the figure below shows how varying $r$ affects reg method 2:



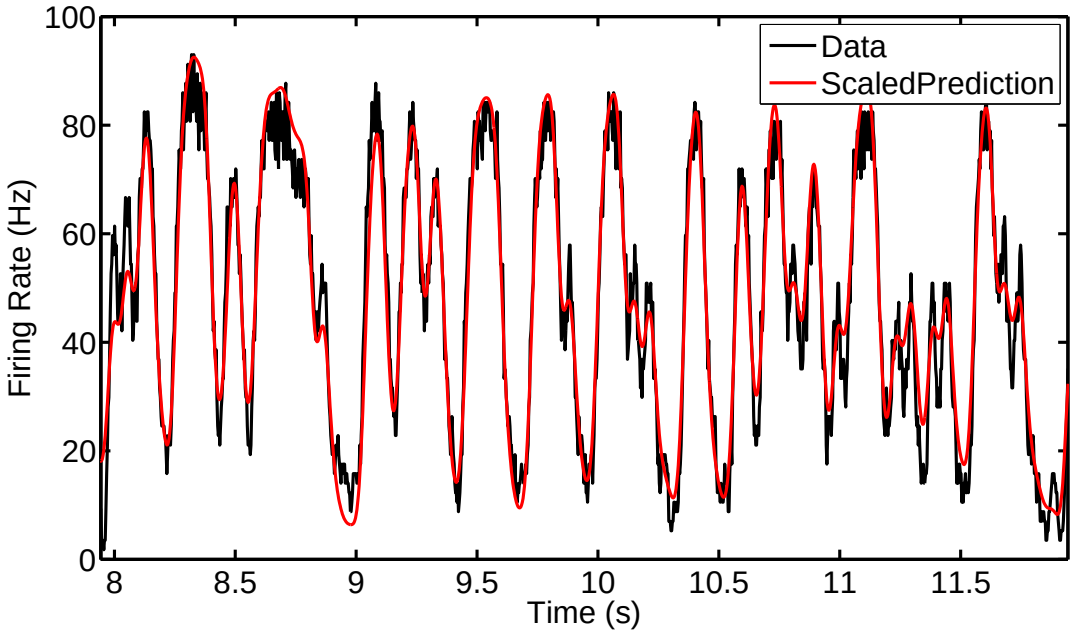

and the best filters look like:




Reg. method 2 leads to a minimum in the value of $S$ at $r = 1$, which means that the mean of the eigenvalues of $C$ is chosen. This seems natural, and produces nice-looking filters with little high-frequency noise on them. The gain, though, is off, but can be corrected post-hoc by scaling the filter:




This filter now has gain of exactly 1. The prediction is pretty good:




The r-square of this prediction is

```
0.9279
```

and the best prediction from all the regularisation factors has a r-square of:

```
0.9446
```